React & JavaScript Interview Preparation Handbook

■ Table of Contents

- React Core Concepts
- JavaScript Advanced Concepts
- UI & Styling
- Performance Optimization

React Core Concepts

Why React over Angular/Vue?

React is unopinionated, lightweight, and focused on UI with a huge ecosystem. Angular is more opinionated and complex, Vue is simpler but has smaller enterprise adoption.

What is Virtual DOM in React?

A lightweight JS object representing the real DOM. React diffs it with the real DOM and only updates changed nodes for efficiency.

Why do we need Redux when Context API exists?

Redux provides structured state management, middleware support, devtools, and better scaling. Context works for small-medium apps but is not optimized for large-scale state logic.

Why are keys required in React lists?

Keys uniquely identify list elements so React can optimize reconciliation without re-rendering the entire list.

Tricky Output Example

function App(){ const [count, setCount] = React.useState(0); setCount(count+1); console.log(count); return {count}; } Output: Infinite re-renders.

JavaScript Advanced Concepts

Difference between let, var, const?

`var` is function-scoped and hoisted, `let` and `const` are block-scoped. `const` cannot be reassigned.

What is Temporal Dead Zone?

The phase between hoisting and initialization for let/const variables where accessing them throws ReferenceError.

Explain closures with example.

Closures let a function access variables from its outer scope even after the outer function returns.

Tricky Output Example

console.log([] + {}); // "[object Object]" console.log({} + []); // 0

UI & Styling

CSS Modules vs Styled Components vs Tailwind

CSS Modules scope CSS per file. Styled Components use CSS-in-JS. Tailwind is utility-first CSS framework.

Responsive design systems in React?

Use CSS Grid, Flexbox, media queries, rem/em units, and libraries like Material UI responsive grid system.

Performance Optimization

useMemo vs useCallback?

useMemo caches computed values; useCallback caches function references to avoid re-renders.

Lazy loading importance?

Loads components only when needed, reducing initial bundle size.

Tricky Output Example

const a = [1,2,3]; const b = [...a]; console.log(a===b); // false